Application Study on Intrusion Detection System Using IRBF

Yichun Peng[1,2,3], Yunpeng Wang[1,3] , Yi Niu[2] , Qiwei Hu[2]
[1]Guangzhou Institute of Geochemistry Chinese Academy of Sciences, Guangzhou, China
Email: yichunpeng678@hotmail.com
[2] City College of Dongguan University of Technology Dongguan, China
[3] University of Chinese Academy of Sciences, Beijing, China
Email: wangyp@gig.ac.cn, ny3388@163.com

***Abstract*—As an active and dynamic security-defense technique, intrusion detection can detect the interior and exterior attacks, and it plays an important role in assuring the network security. Based on immune recognition algorithm, a Radial Basis Function (RBF) neural network learning algorithm was studied. In this algorithm, the input data is regarded as antigens and antibodies are regarded as the hidden layer centers, the weights of the output layer are determined by adopting the Recursive Least Square method, which can improve convergence speed and precision of the RBF neural network, using Snort to establish innate antibody and using negative selection algorithm to generate detectors. This algorithm was applied to Intrusion Detection Systems. Theory and experiment show that this algorithm has better ability in intrusion detection, and can be used to improve the efficiency of intrusion detection, and reduce the false alarm rate.**

***Index Terms*—Intrusion Detection; Radial Basis Function Neural Network; Clonal Selection; Immune Algorithm; Snort; Negative Selection**

I. Introduction

Network, especially Internet, has brought unprecedented chances and challenges to the society in the 21st century. On one hand, the normal running of network has brought great progresses and wealth to the society. On the other hand, it has caused unexpected disasters and losses by the insecurity of network. The relationship between security threats and ensured safety is just like the spear and the shield. Nowadays, in order to deal with all kinds of infinite changing attacks, people have taken a variety of anti-attack means, among which intrusion detection system (IDS) is the second security gate behind the firewall. IDS can be used to detect all kinds of intrusion behaviors, active intercept, and dynamic reacts to the vicious intrusions before the network system is jeopardized. However, the promotion of network bandwidth, the diversity of attacking forms, the diversification of attacking means, and intelligence of attacking technologies have increased the difficulty of intrusion detection, which has greatly reduced the practicability of the intrusion detection system. The disadvantages include high rates of incorrectness of detection intrusion, high rates of false positive, and bad activations. The following are the biggest disadvantages:1) the high missing rate of detection 2) the high false detecting rate 3) poor autonomous and less intelligent detection 4) mainly by human activities not automatically action after detecting invasions. In order to overcome the deficiencies of the existing IDS, in this paper, Radial Basis Function (RBF) neural network and Immune Algorithm (IA) are combined to form a kind of Immune Radial Basis RBF (IRBF) network training Algorithm, which can not only distinguish normal and abnormal data in the network, but also diagnose invasion types, take special precautions and further respond to specific invasions as well. The experimental results showed that IRBF based intrusion detection technology can accurately distinguish and identify a variety of parallel invasion. It has faster processing speed, enough fault-tolerant and the powerful ability of self-learning and self-adjusting. Therefore, intrusion detection systems can be a truly intelligent security gate in network.

II. Related Technologies

A. RBF Neural Network

RBF neural network is a kind of local approximation neural network, which has very strong approximation ability, classification ability and learning speed. Its working principle is to put the network as the approximation of unknown function, which means any functions can be expressed as weighted sum for a set of basis functions. That is to say, it chooses transfer functions of each hidden layer's neurons to form a set of basis function which approximate the unknown function. The structure of RBF neural network is shown in Figure 1. It is a three-layer feed forward neural network: input layer, hidden layer and output layer, and the number of their respective units are m, q, p.

This work was supported in part by a grant from the Scientific and Technological Projects of Guangdong (No. 2009B010800042), and in part by Projects of Science and Technology of Dongguan(No. 201110825100119).

doi:10.4304/jsw.9.1.177-183

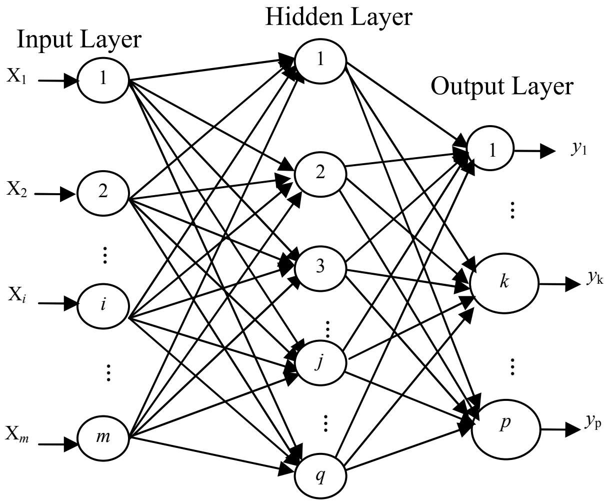


Figure 1. The Structure of RBF Neural Network

The hidden layer of RBF neural network chooses the function of Gauss as the basis function, which sets the input of the input layer as $X = (x_1, x_2, ..., x_j, ..., x_n)$, but the actual output is $Y = (y_1, y_2, ..., y_k, ..., y_p)$., Input layer is to achieve non-linear mapping of $X \to R_i(x)$, output layer is to realize linear mapping of $R_i(x) \to y_k$, network output of the K-th neuron of output layer is:

$$\hat{y}_k = \sum_{i=1}^{m} \omega_{ik} R_i(x) \qquad (k = 1,2,..., p) \tag{1}$$

Where, n is the input layer nodes, m is the hidden layer nodes, and p is the output layer nodes; ω_{ik} is the connection weight between the i-th neuron of hidden layer and the K-th neuron of output layer. $R_i(x)$ is an action function of the i-th neuron of hidden layer, just as formula (2) below:

$$R_i(x) = \exp(-\|x - c_i\|^2 / 2\sigma_i^2), (i = 1,2, \text{L}, m) \tag{2}$$

Where, x is an n-dimensional input vector, ci is the center of the k-th basis function, the same dimension vector as x, σi, which determines the width of the basis function around the center point, is the i-th perception variables; m is the number of perception unit (hidden node number). $\|x - c_i\|$ is the norm of vector x-ci, and it usually indicates the distance between x and ci; $R_i(x)$ has a unique maximum value in ci, with the increasing of $\|\text{x} - \text{c}_\text{i}\|$, $R_i(x)$ rapidly decays to zero. For a given input, only a small portion near the centre of x is activated. When the cluster center ci of RBF network and weights ω_{ik} are determined, we can obtain the network output value according to a certain input.

It is critical to select the center ci of RBF neural network. The following are the two main ways: according to the experience and using the clustering method (such as K means clustering method). However, these two ways are more difficult to achieve the network global optimal value of the center and width of basic function. In this paper, we use immune recognition algorithm which is based on clonal selection to determine the center of RBF neural network.

B. Immune Algorithm

The concept of modern immunity refers to the function that the body distinguishes between self and non-self, even excludes non-self, and the purpose is to maintain its own physiological balance and stability. Immune algorithm is a new kind of intelligent learning algorithm which simulates the biology immunity systems; it is one of the main content of the Artificial Immune System (AIS). Immune algorithm has a good system response, dynamic and autonomy, and has the strong ability to maintain self-balancing when the system is interfered. In addition, immune algorithm also simulates some special functions of immune system such as learning, memory and recognition. It has the very strong ability to classify the pattern, especially for multi-modal problem analysis, processing and solving, which exhibits a higher intelligence and robustness. Immune algorithm has been applied to a variety of single target, multi-objective optimization and engineering optimization, for example, automatic control, abnormal and fault diagnosis, the robot behavior simulation, robot control, network intrusion detection, neural network design and other fields, and has shown more excellent performance and efficiency. In addition, immune algorithm has been applied in pattern recognition, image recognition, design optimization, data mining, information processing, associative memory, bank identification for mortgage fraud, etc.

III. Research on Intrusion Detection System Based on IRBF Neural Network Learning Algorithm

In this paper, the IRBF neural network training algorithm is that it uses immune recognition algorithm which is based on clonal selection to determine the data center of hidden layer of RBF neural network, and then uses the formula (2) to calculate the output of hidden layer after the center of hidden layer is determined. Furthermore, it uses the least squares method to calculate the weighted between hidden layer nodes and output nodes. Finally, we can get the final output of the RBF neural network.

A. Uses Immune Recognition Algorithm Which is Based on Clonal Selection to Determine the Data Center of Hidden Layer of RBF Neural Network

In the course of development of immune algorithm, immune recognition algorithm based on clonal selection has been investigated by many scholars. It is an evolutionary algorithm which simulates the learning process of the biological immune systems. its essence is Darwinian selection and mutation theory, using the strategy of antibody set population updating.

The input data as antigen, RBF network of the hidden layer center is corresponding to antibody, using this algorithm to get the diversity of antibody memory set as the hidden layer data center. The algorithm's steps are as follows:

(1) Initializing. Input antigen and determining the size of the initial population N, the total number of clone M, randomly generated N antibodies that constitute the initial antibody population A_0, which is defined as the problem of possible solutions (or random solution).

(2) Calculate affinity. Calculate the affinity of each antibody in the A_0, and A_0 will be decomposed into A_m and A_r, wherein, A_m represents an antibody memory set in which the affinity of antibody is higher, A_r represents the remainder of the antibody set. The computation formula of the affinity is as follows:

The matching degree between antigen x_i and antibody c_j is called affinity degree:

$$a_{ij} = \frac{1}{1+\left\|x_i - c_j\right\|} \qquad (3)$$

When $x_i = c_j$, affinity degree a_{ij}=1 is maximum;

The matching degree between antigen c_i and c_j is called similarity degree:

$$s_{ij} = \frac{1}{1+\left\|c_i - c_j\right\|} \qquad (4)$$

When c_i=c_j, similarity degree s_{ij}=1 is maximum;

(3) Cloning. To choose k antibodies whose affinity degree is the highest for cloning, the number of cloning is proportional to the affinity degree of antibody, that is, the higher affinity antibodies have a higher chance to be searched.

(4) Variation. Each clone cell is mutated according to its affinity mutation operator, the mutation bit number is inversely proportional to the affinity of its antigen, to get cloning set C_n, cloning inhibition operator acting on C_n and to get the cloning set C_n^*, and to calculate the affinity of each new antibody and antigen.

(5) Excellent antibodies screening. In C_n^*, if the highest affinity antibody whose affinity is even higher than the affinity of its parent antibody, then using it to replace the original antibody, and form a new memory set D_m.

(6) Clone inhibition. Calculating the similarity degree among those antibodies, the higher the similarity degree of antibody, the greater the antibiotic activity, . Selecting the inhibition threshold and exclude those high similarity degree antibody according to the formula below:

$$t_s = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} \mathrm{s}_{\mathrm{ij}}}{M(M-1)} \times 2 \qquad (5)$$

Where, M represents the total number of antibody in the current antibody set.

(7) Generate memory set. Detecting all antigen whether they have learned completely, if yes, then merging all antibody memory sets which including all antibodies generated by antigen; otherwise, turn to step 2), restart.

(8) Immune depression. Calculating the similarity degree among each memory cell, and excluding those high similarity degree antibody according to formula (5) .

(9) Checking whether it reaches the maximum evolution generation. If so, the current memory set is the optimal solution of the problem, then to output results and end, otherwise, turn to step 2), restart.

B. Least Squares Recursive Method to Adjust Weights

There are three methods to calculate the weight between the hidden layer nodes and the output nodes, which are negative gradient descent direction of minimum variance method, recursive least squares method (RLS) and mirrored the least squares method. Among them, the RLS is often used because it has good global convergences. Define the objective function:

$$j(t) = \sum_{\mathrm{p}=1}^{\mathrm{L}} \mathrm{E_p(t)} = \frac{1}{2}\sum_{\mathrm{p}=1}^{\mathrm{L}} \lambda(\mathrm{p})[\mathrm{d_p(t)} - \mathrm{y_p(t)}]^2 \qquad (6)$$

Where, L is the length of the sample, $\lambda(\mathrm{p})$ is the weighting factor(0< $\lambda(\mathrm{p})$ <1), $\mathrm{d_p}$ is the target output sample, $\mathrm{y_p}$ is the actual output sample.

Through $\frac{\partial \mathrm{j(t)}}{\partial \mathrm{w}} = 0$ to calculate W, which makes $j(t)$ get minimum value, then, this w is the required weight.

$$\begin{aligned} w_p(t) &= w_p(t-1) + k(t)[d_p(t) - q_p^T(t)w_p(t-1)] \\ k(t) &= p(t-1)q_p(t)[q_p^T(t)p(t-1)q_p(t) + \frac{1}{\lambda(p)}]^{-1} \\ p(t) &= [L - k(t)q_p^T(t)]p(t-1) \end{aligned} \qquad (7)$$

Where, $q_p(t) = [q_{1p}(t), q_{2p}(t), \ldots, q_{Lp}(t)]^{\mathrm{T}}$, L is the number of hidden layer nodes.

Because the weight only exists between the hidden nodes and the output nodes, the computational requirements for the RBF network training are very less. Neural network is trained to generate the appropriate structures and parameters, and will be able to detect the network data in intrusion detection.

C. IRBF based on Intrusion Detection Model

The structure of intrusion detection model based on IRBF is shown in Figure 2:

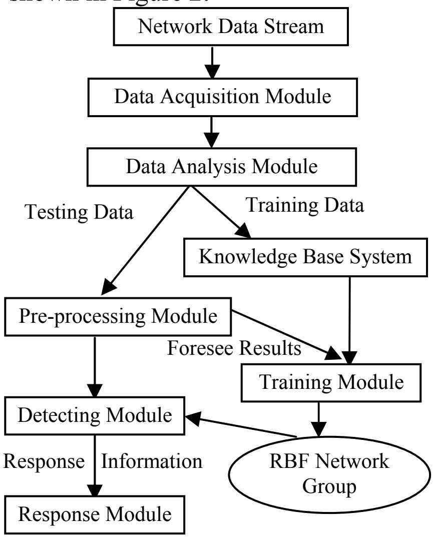


Figure 2. The Structure of Intrusion Detection Model Based on IRBF

Each module is described as follows:

(1) Data acquisition modules. It's responsible for

grasping network packets, and sends them into data analysis module.

(2) Data analysis module. Data analysis module can be snort, TCP dump or development packet protocol analysis program, its function is packet filtering and classification, then it obtains the required data header information, and stores it. Firstly, preliminary filtering, checking the format of IP packets, restructuring if it is subdivision. Secondly, distinguishing IP packet, UDP packet, or ICMP packet event, and then calling and semantic analyzing different analyzer program segment according to the different protocol type of the packet. Finally, sending the information of the packet which meets the requirement to the pre-processing module.

(3) Pre-processing module. In the process of generating detector, the received training samples will be transformed into binary format by TCPdump For detecting process, the received pocket header information, which will be transformed into binary format by TCPdump, and as the input of RBF neural network, the processed data can also be directly sent into the detector module, but if attack types of the current data has been identified, then the processed data can be send into the training module for RBF neural network training, in order to achieve the real-time monitoring and continuous self-learning.

(4) Knowledge base system. Knowledge base system can be replaced with a simply including abnormal behavior rules set and it is used to decide what kind of packets and session will be monitored and accessed. The existence of these rules has two functions. One corresponds to the congenital antibody of biological immune system, and can accurately detect some known intrusions; the other puts forth some limit condition when the detector is generated to reduce the generating time and the occupied space of the detector.

(5) Training module. Combined with knowledge base system and foresee results, using the above algorithm to train the pre-processed data and generate RBF network group.

(6) RBF network group. It is the core part of intrusion detection, and used for detecting data and outputting the result. RBF network group pertinently enters the detecting module to detect data, and at the same time, it is also continuously updated, with the arrival of the new learning data and detecting continuously, the number and internal structure of the RBF network group will also be changed.

(7) Detecting module. Containing the most affinity RBF network relative to the data waiting for detecting, final discriminating the output of the RBF network group, and sending the result to the response module for processing

(8) Response module. Determining the intrusion behavior and alarm information to notify the network administrator or take corresponding measures, such as cutting off connections, tracking the attacker, etc.

Among them, the detector module plays a vital role in the intrusion detection system, and at the same time, without response module, the intrusion detection system will lose the value of its existence. The following are the further study on these two modules.

1) Generation of the Detector

The human immune system can be divided into innate and acquired immune by the immune response. Therefore, this system also be designed two kinds of detector: one kind is innate detector, is to establish detection rules based on expert knowledge to realize the same known intrusion detection, the other kind is acquired detector, which is to be established by negative selection to achieve the unknown intrusion detection.

While the system is running, its antibody vaccine is produced by the antibody gene library which includes expert antibody library and adaptive antibody gene library, among which:

Expert antibody library consists of created rules, which is equivalent to the body's innate immunity. As the body's immune system can't resist all allogenic attacks, networks may also contain some new form of attacks, which may escape detection range of the detection system, in this case, establishing a new detection rule based on this attack to implement the intrusion detection. To establish an expert antibody library not only can improve some known intrusion detection efficiency and accuracy bat also can decrease the expenditure of time and space to generate adaptive antibody library.

The adaptive antibody library is a new antibody library which is formed by the immune genetic computing mechanism. The antibody of adaptive antibody library was formed after the field (i.e. the antibody gene) which represents the characteristics of the invasion in the network data packet header to be encoded. And each antibody has an adaptive value A, if some antibody (detector) has successfully detected corresponding intrusion behavior, then A value is added 1; Conversely, if some detector has not detected the invasion for long time (survival period), then the A value is misused 1.When the value of A is less than 0, the detector will be deleted from the adaptive antibody library, which is equivalent to the biological cell death. This mechanism can ensure that the adaptive antibody library saves its most active antibody, which is also consistent with the biology of natural selection, namely the principle of survival of the fittest. In the intrusion detection system, which generates candidate antibody library after the existing alien mode (namely antigen) has been mutated, and there is a negative selection on each new candidate antibody with existing alien pattern. If they are completely matched then new candidate antibody will be deleted, the last remaining effective antibody is stored in the adaptive antibody library, and is used for the detector to detect intrusion.

i.The Establishment of Innate Antibody

The establishment of innate antibody is based on misuse intrusion detection method. Here, we adopted Snort-based intrusion behavior description method, which is simple, easy to implement, detection rapidly and can describe the vast majority of intrusion behavior.

Snort stored all known attacks in the form of rules in the rule library. Each rule is composed of the rule header and the rule option. Rule header is corresponding to the

Rule Tree Node (RTN), including action, protocol, source (destination) address, port and data flow direction; Rule option is corresponding to the Optional Tree Node (OTN), including alarm information (MSG), matching content options. It is not an essential part of the rule and it is only used to define and collect specific characteristics of specific data packets. When Snort initialize and parse rules, TCP, UDP, ICMP and IP four different rules tree are established respectively, each rule tree contains independent 3D linked list: RTN (rule head), OTN (rule option) and point matching function pointer.

Here is an example to illustrate the composition of the rules:

alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg:"mounted access";)

Description: data packets of 111 port of any host that uses the TCP protocol to connect to 192.168.1.0/24 network, if in which a binary data “00 01 86 a5” has appeared, then the system will issue a warning message” mounted access”.

In this rule: "alert TCP any any->192.168.1.0/24 111" is the rule header; "content:’|00 0186 a5|’;msg:’mounted access’;" is the rule option, in front of colon phrases is called the option keyword. A rule only to be executed when different parts of it must be satisfied simultaneously which is equivalent to the "and" operation, and among all rules is equivalent to an "or" operation in a same rule database file.

The first part of rule header is action, which indicates what should be done when it finds the data pocket that is suitable for its conditions. There are five predefined action: Alert, Log, Pass Activate and Dynamic, and can also customize action; The second part is protocol, which shows what type of pocket will be compared with the rule. At present, Snort supports the following protocols: TCP, the IP, UDP, and ICMP; The third part is data packet's source and destination IP address and port, behind the IP address, it specifies the network mask, for example, /16 specify the class B network, /24 specify the class C network, and /32 specify a particular host, the port number can be specified in several ways: "ANY", digital, ":" (range) and "!" (not), where "ANY" specify any port, number specifies a single port, such as 80 for HTTP, 23 for telnet, etc, between these two address and port is direction section, which determine the source and destination, "->" indicates a direction from left to right, "<-" indicates a direction from right to left , and "<>" indicates the rule will be used in both directions. You can use it when the system must monitor server and client simultaneously, for example, monitoring the data flow of POP or Telnet server that comes and goes.

Rule option may include more than one option, there use ";" to separate different options, and use logic operator "AND" to express it. Option consisted of keywords and parameters, each keyword and its parameters using ":" to separate. At present, those following keywords can be explained: msg,logto, ttl, id, dsize, content, offset, depth, nocase, flags, seq, ack, itype, icode, session, icmp_id, icmp_seq, ipoption, rpc and resp, etc.

ii.The Formation of Adaptive Antibody

Gene and encoding are very important in the intrusion detection system based on biological immune. According to expert the knowledge and experience, we have extracted some intrusion detection features from all fields of network data packet as the gene that will be encoded and be generated antibody (namely detector). Since the operation must be accomplished by reorganization, selection and quantitative calculation of individuals with a certain structure form in the population, it needs a direct digital representation, namely, encoding. Encoding refers to a process that images phenotype into genotype, among them, phenotype refers to readable rules directly received by joining records, genotype means an internal form in proceeding antibody evolution, negative selection and clonal selection, the corresponding relationship between the phenotype and the genotype as shown in Table I.

TABLE I.
THE CORRESPONDING RELATIONSHIP BETWEEN THE PHENOTYPE AND THE GENOTYPE

The phenotype of detector	*ScrIP DesIP ProType Scrport Desport flag …*
The genotype of detector	110011… 011111000… 00000110…00101000

In the paper, we have consulted the principle of using negative selection algorithm to generate detectors, which was proposed by Stephanie Forrest group of Department of Computer at New Mexico University, USA. This principle is shown as in figure 3.

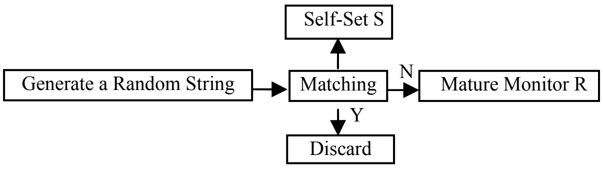


Figure 3. The Generation of Effective monitor

The model was originally used to generate detector set for detecting viruses, and achieve success, later, to be applied in the UNIX operating system which can detect several intrusion, such as sendmail, lpr, however, for the generation of the detector in intrusion detection system, using purely random string (that is, the candidate antibody) to produce effective antibody which is low-efficiency and time consuming. In this paper, we make some improvements to this method. Firstly, encoding the known intrusion patterns, then evolving the genetic operator. Finally, generating candidate antibody library. The steps of algorithm are as follows:

Step 1) Generating self set and non-self set, and encoding.

Step 2) Evolving non-self set through mutation, generating the initial candidate antibody library.

Step 3) Detecting the candidate antibody with expert rules, if in accordance with the rule then delete it, otherwise, turn to Step 4.

Step 4) Matching the candidate antibody generated by mutation with the original antigen, if they are matching then delete the candidate antibody, otherwise, turn to Step 5.

Step 5) Matching those remaining candidate antibodies after selected by Steps 2 and Step 3 with normal set, if they are matching then delete the candidate antibody, if not, the candidate antibody become mature antibody, and will be used to detect intrusion behavior.

Compared to the original algorithm, this algorithm has some advantages as follows: the newly generated candidate detector is based on the existing abnormal mode based, and is not an impossible mode. Thus, it can make the number of candidate detector not need so much like the original algorithm, and these candidate detectors are effective, which will be helpful for protecting the detection efficiency and saving storage space.

2) *Intrusion Response and Confrontation*

The most simple automatic intrusion response of intrusion detection system is to inform: when the system detects intrusion occurs. It will send an E-mail or a message to the administrator. An initiative response is to prevent the ongoing attacks, make the attacker unable to continue to access. For example, truncating the connection between the attacker and the target host through injecting reset datagram, restricting the access of an intruder through updating the configuration of firewall, and so on; A more initiative response is to counterattack the attacker, but this method is very dangerous, it may affect innocent users in the network, but also is illegal. Automatic response is the cheapest and the easiest response way, as long as can be wise and carefully implemented. It is still relatively safe. However, it has two problems: for one thing，since the system has the potential to create false alarm, there may response to a network node which never attacks us by mistake; for another, if an attacker determines our system with automatic response, he might take advantage of this to attack us, for example, he may connect two network nodes with automatic response intrusion detection system to establish a feedback loop which is equivalent to echo-charged, then makes address spoofing attack to those two nodes.

Take different response mechanisms for different types of "non-self": for connection-oriented protocol (such as illegal TCP connection), directly reset it; for non-connection-oriented protocols (such as illegal UDP data), alarm and timely notify the system administrator. If it is found that other unknown protocol (e.g., the system's own non-standard protocol) can also alarm and notify the system administrator, the administrator check system alarm and log, if it is the system normal data then confirm its legality in safety rules and become a member of "self" set, otherwise, there need timely adjustment of the firewall rules (e.g., using linkage method) to perfect object security rules and trace intrusions.

The process of alarm decision-making is as follows:

The response module receives alarm information from the detector module, firstly, alarm information will be analyzed locally, if it can’t be determine, this response module requests other node’s response module to determine local analysis by requisition-coordination, which is based on cooperative mechanism of biological immune. Its basic principle is: if more than a certain number of detector modules detect the behavior and administrators also confirm that it is intrusion behavior, the behavior is an intrusion behavior; Then, in determining the alarm is a kind of intrusion behavior, response module makes general response, but also converts alarm into memory antibody, the administrator can also generates memory detector through the features generator of the detection module; Finally, the response module distributes memory detector to other nodes, so that other nodes can rapidly response to the same invasion in the future, which draw lessons from biological immune secondary response mechanism.

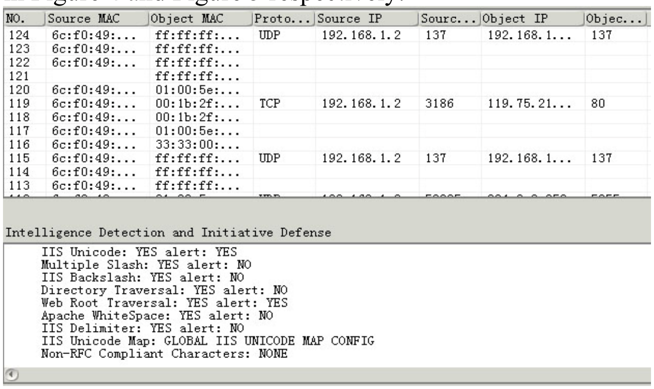
IV. Experimental Results and Analysis

The experiment adopts KDD Cup 1999 data set released by the Lincoln Laboratory of the United States. This data set consists of 4 categories (PROBE, DoS, U2R, R2L) , and 39 attack types for a total of 5000000 records; it also provides a 10 percent training subset and testing subset, there are 22 of attacks in the training subset and 17 types remained in the testing set. The purpose of this design is to test the generalization ability of the classifier model. The ability to detect the unknown attack types is an important index to evaluate whether the intrusion detection system is good or not

As to selecting samples, selecting Dos and Normal types data, and only choosing R2L, Probe types data records mixed with R2L type data. There are 494021 records in the selected 10 percent data subset, which includes 97278 normal data records and 396473 abnormal data records. The proportion of normal data to abnormal data is 9:1, which is in conformity with general rules of intrusion detection.

We selected 11 representative vector elements in RBF neural network’s input layer: source IP address, destination IP address, the IP packet flag, the total length of the IP packet, IP header length, protocol code, source port number, destination port number, TCP flags, TCP window size, and the first byte of the data segment. These data are input into RBF network after they have been normalized.

In order to make “own” library development perfect, firstly, the intrusion detection system should learn in safety network flow. The system will gradually produce antibodies in the training process. Experiments are made on each type of attack, and each type uses 10 records. There are two ways for experiments to test: real-time testing and off-line testing. The results (part) are shown in Figure 4 and Figure 5 respectively:

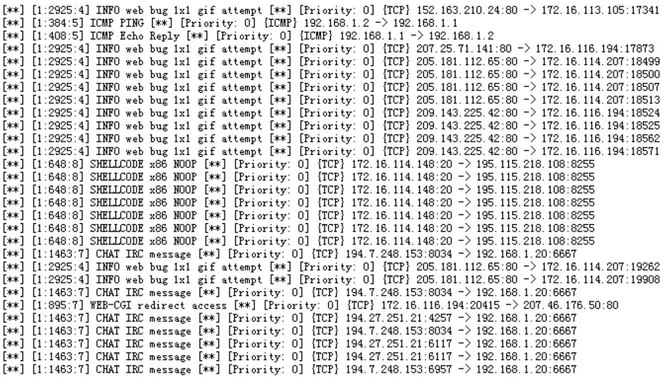


NO.	Source MAC	Object MAC	Proto...	Source IP	Sourc...	Object IP	Objec...
124	6c:f0:49:...	ff:ff:ff:...	UDP	192.168.1.2	137	192.168.1...	137
123	6c:f0:49:...	ff:ff:ff:...					
122	6c:f0:49:...	ff:ff:ff:...					
121		ff:ff:ff:...					
120	6c:f0:49:...	01:00:5e:...					
119	6c:f0:49:...	00:1b:2f:...	TCP	192.168.1.2	3186	119.75.21...	80
118	6c:f0:49:...	00:1b:2f:...					
117	6c:f0:49:...	01:00:5e:...					
116	6c:f0:49:...	33:33:00:...					
115	6c:f0:49:...	ff:ff:ff:...	UDP	192.168.1.2	137	192.168.1...	137
114	6c:f0:49:...	ff:ff:ff:...					
113	6c:f0:49:...	ff:ff:ff:...					

Intelligence Detection and Initiative Defense

```
    IIS Unicode: YES alert: YES
    Multiple Slash: YES alert: NO
    IIS Backslash: YES alert: NO
    Directory Traversal: YES alert: NO
    Web Root Traversal: YES alert: YES
    Apache WhiteSpace: YES alert: NO
    IIS Delimiter: YES alert: NO
    IIS Unicode Map: GLOBAL IIS UNICODE MAP CONFIG
    Non-RFC Compliant Characters: NONE
```

Figure 4. The Result of Real-time Testing (Part)



```
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 152.163.210.24:80 -> 172.16.113.105:17341
[**] [1:384:5] ICMP PING [**] [Priority: 0] {ICMP} 192.168.1.2 -> 192.168.1.1
[**] [1:408:5] ICMP Echo Reply [**] [Priority: 0] {ICMP} 192.168.1.1 -> 192.168.1.2
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 207.25.71.141:80 -> 172.16.116.194:17873
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 205.181.112.65:80 -> 172.16.114.207:18499
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 205.181.112.65:80 -> 172.16.114.207:18500
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 205.181.112.65:80 -> 172.16.114.207:18507
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 205.181.112.65:80 -> 172.16.114.207:18513
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 209.143.225.42:80 -> 172.16.116.194:18524
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 209.143.225.42:80 -> 172.16.116.194:18525
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 209.143.225.42:80 -> 172.16.116.194:18562
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 209.143.225.42:80 -> 172.16.116.194:18571
[**] [1:648:8] SHELLCODE x86 NOOP [**] [Priority: 0] {TCP} 172.16.114.148:20 -> 195.115.218.108:8255
[**] [1:648:8] SHELLCODE x86 NOOP [**] [Priority: 0] {TCP} 172.16.114.148:20 -> 195.115.218.108:8255
[**] [1:648:8] SHELLCODE x86 NOOP [**] [Priority: 0] {TCP} 172.16.114.148:20 -> 195.115.218.108:8255
[**] [1:648:8] SHELLCODE x86 NOOP [**] [Priority: 0] {TCP} 172.16.114.148:20 -> 195.115.218.108:8255
[**] [1:648:8] SHELLCODE x86 NOOP [**] [Priority: 0] {TCP} 172.16.114.148:20 -> 195.115.218.108:8255
[**] [1:648:8] SHELLCODE x86 NOOP [**] [Priority: 0] {TCP} 172.16.114.148:20 -> 195.115.218.108:8255
[**] [1:648:8] SHELLCODE x86 NOOP [**] [Priority: 0] {TCP} 172.16.114.148:20 -> 195.115.218.108:8255
[**] [1:1463:7] CHAT IRC message [**] [Priority: 0] {TCP} 194.7.248.153:8034 -> 192.168.1.20:6667
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 205.181.112.65:80 -> 172.16.114.207:19262
[**] [1:2925:4] INFO web bug 1x1 gif attempt [**] [Priority: 0] {TCP} 205.181.112.65:80 -> 172.16.114.207:19908
[**] [1:1463:7] CHAT IRC message [**] [Priority: 0] {TCP} 194.7.248.153:8034 -> 192.168.1.20:6667
[**] [1:895:7] WEB-CGI redirect access [**] [Priority: 0] {TCP} 172.16.116.194:20415 -> 207.46.176.50:80
[**] [1:1463:7] CHAT IRC message [**] [Priority: 0] {TCP} 194.27.251.21:4257 -> 192.168.1.20:6667
[**] [1:1463:7] CHAT IRC message [**] [Priority: 0] {TCP} 194.7.248.153:8034 -> 192.168.1.20:6667
[**] [1:1463:7] CHAT IRC message [**] [Priority: 0] {TCP} 194.27.251.21:6117 -> 192.168.1.20:6667
[**] [1:1463:7] CHAT IRC message [**] [Priority: 0] {TCP} 194.27.251.21:6117 -> 192.168.1.20:6667
[**] [1:1463:7] CHAT IRC message [**] [Priority: 0] {TCP} 194.7.248.153:6957 -> 192.168.1.20:6667
```

Figure 5. The Result of Off-line Testing (Part)

In the last experiment, we have sampled all types of data records by interval sampling, and observed the detection precision as shown in Table II in four experiments:

TABLE II.
THE OFF-LINE DETECTION RESULT

Sample Type	*Detection Precision*
Dos+Normal	98.6%
R2L+Normal	69.5%
Probe+Normal	75.7%
All Types	83.7%

The experiments showed that it is not only able to distinguish between “self” and “non-self” network data, but also to identify those four types of known intrusion: DoS, R2L, U2R, and probing. However, nowadays no intrusion detection system can find all the invasion, and still exist false positive and false negative. IRBF algorithm also contains such defects.

V. Conclusions

This paper presents an intrusion detection system model based on IRBF neural network learning algorithm, and the model determines the center of RBF neural network by means of cloning, mutation and suppression immune algorithm, and uses the recursive least square method to adjust the weight between hidden layer and output layer of RBF network. Moreover, the model has advantages of less computation, faster convergence speed, high precision, and good real-time performance. The model is applied to intrusion detection system, and the experiments showed that it can not only detect the known intrusion, but also detect unknown intrusion to some extent, which reduces the rate of false positives and false negative rate of traditional intrusion detection systems, and improves the system learning efficiency and intelligence.

References

[1] Li Xin-yu, Zhou Tie-jun, “Research of Intrusion Detection Optimization Algorithm based on RBF Neural Network,” *Computer Security*, no.4, 2011,pp.29-32.

[2] Liu Dihua, Yu Bin, Wang Xiaofen, “Research on Intrusion Detection Model based on RBF Neural Network,” *Network Security Technology & Application*, no.12,2008,pp.36-38.

[3] Huang Yuanjiang, “Prediction of debris flow based on IRBF neural network,” *Journal of Central South University of Forestry & Technology*, no.3,2010,pp.159-163.

[4] Deng Guanghui, Jing Dongxing, Ye Jixiang, “Emotion Recognition of Speech Signals Based on the Immune Radial Basis Function Networks,” *Computer Engineering & Science*,no.9,2009,pp.153-159.

[5] Qiu Chuchu, You Dade,Ma Ye, Wang Weiyan, “Prediction of carrier UAV based on IRBF neural network,”, *Ship Science and Technology*,2011(11):109-111.

[6] Bao Wensheng, Liu Xiaogang, “Structure Optimization Algorithm of RBF Neural Networks based on Adaptive Genetic Algorithem,” *Journal of Shandong Normal University(Natural Science)*,no.3,2007,pp.37-39.

[7] Yang Shu-yuan , Jiao Li-cheng, Liu Fang, “An immune RBF neural network MUD method,” *Journal of XIDIAN University (Natural Science)*,no.4,2004,pp.209-213.

[8] Niu Yi, Zhang Quanju, Zheng QiLun, Peng Hong, “Security Operation Center Based on Immune System,” *Workshops International Conference on Computational Intellignece and Security. Harbin,China: IEEE Computer Seciety Press*, no.11,2007,pp.97-103.

[9] Niu Yi, Zheng QiLun, Peng Hong, “Security Operation Center Design Based on Radial Basis Function Neural Network,” *Dynamics of Continuous, Discrete and Impulsive Systems*, no3,2007,pp.1133-1140

[10] Paul K Harmer, Paul D Williams, Gregg H Gunsch, Gray B Lamont, “An Artificial Immune System Architecture for Computer Security Applications,” *IEEE Transactions on Evolutionary Computation*, no.3,2002,pp.252-280.

[11] Lei Wang, Beat Hirsbrunner, “Immune Mechanism Based Computer Security Design,” *Proceeding of the First International Conference on Machine Learning and Cybernetics*, no.11,2002, pp.1887-1893.

[12] Hamid Reza Golmakani, Elnaz Jalilipour Alishah, “Portfolio Selection Using An Artificial Immune System,” *IEEE Congress on Information Reuse and Integrations*, vol.28, no.3, 2008, pp.65-72.